

## METHOD AND APPARATUS FOR PRINTING COMPUTER GENERATED IMAGES

### Technical Field of the Invention

The present invention relates generally to printing of computer generated images and, in particular, to utilising regional segmentation and an improved modular extensible  
5 electronics based architecture for printing of computer generated images.

### Background Art

Printing devices are well-known for processing encoded page layout information to form a printed page. The page layout information is first formed into decoded and  
10 uncompressed page image data. The page image data is then sent to a raster scanned device, which prints discrete picture elements (pixels) at fixed physical locations on an output medium such as paper. Typically, the pixels have different colours, different intensities and possibly other variable parameters.

In many types of printing devices a high level description of image data to be  
15 printed is processed by a raster image processor (RIP) which converts the description into data held in memory for subsequent output to the raster scanned device. The high level description typically takes the form of a page description language (PDL) specification, which is an interchange standard for representing documents electronically within a system and ultimately printing those documents. A PDL file describes, among other  
20 things, how images are to be rendered on each page, what font is to be employed and how the pages are to be assembled. In other printer types, the high level description takes the form of bitmap image data which is transcoded into a format suitable for the raster output device for printing. Such printing devices usually containing a command language which enable transfer of said image data.

All the aforementioned printer types use memory for storing intermediate print data and some raster image processor (RIP) to format data into a representation suitable for the print output device.

Those skilled in the art of design of printer devices will appreciate that printing at high resolutions requires the generation of a large amount of raster output data and a correspondingly high amount of computational processing, that being both for rasterisation of a page description and colour conversion of rendered image data for colour output devices.

Game consoles typically comprise components for providing fast rendering of images on a display screen. Such game consoles typically do not have printing capabilities. Prior art game consoles with printing capabilities are only able to print to a resolution up to that of the display screen.

Furthermore, a raster image processor in such a game console is not intended to be used for printing, and in particular at resolutions higher than that of the display screen, as the raster image processor is incapable of generating the large amount of raster output data required for such a higher resolution print. This is due to limitations in processing capacity and raster image processor memory in such systems.

### **Summary of the Invention**

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

According to a first aspect of the invention, there is provided a method of rendering an output image on an output device at a first resolution using a rendering device limited to rendering to a second resolution, said second resolution being lower than said first resolution, said method comprising the steps of:

segmenting said output image into a plurality of sub-areas, wherein each sub-area is capable of being rendered by said rendering device at said second resolution;

rendering of each sub-area by said rendering device at a resolution not more than said second resolution;

5 combining said rendered sub-areas to form a band of said output image; and  
outputting said band on said output device at said first resolution.

According to a second aspect of the invention, there is provided a method of rendering an output image on a raster scanned device at a first resolution, said method comprising the steps of:

10 segmenting said output image into a plurality of sub-areas, said sub-areas having a second resolution and said second resolution being lower than said first resolution, wherein longitudinal sub-areas form bands;

sequentially rendering image data for said sub-areas at a second resolution and by a rendering device limited to said second resolution;

15 upon each of said sub-areas being rendered, transferring said image data to a band buffer;

colour converting said image data to form raster data suitable for said raster scanned device; and

upon all sub-areas of a current band being transferred into said band buffer,  
20 transferring said raster data of said current band to said raster scanned device.

According to a third aspect of the invention, there is provided an apparatus for rendering an output image on an output device at a first resolution, said apparatus comprising:

a first processor for segmenting said output image into a plurality of sub-areas,  
25 each sub-area having a second resolution and said second resolution being lower than said first resolution;

a second processor for rendering each sub-area at said second resolution, said second processor having a resolution limit sufficient for rendering said second resolution, but less than said first resolution,;

means for combining said rendered sub-areas to form a band of said output image;

5 and

means for outputting said band on said output device at said first resolution.

According to a fourth aspect of the invention, there is provided an apparatus for rendering an output image on a raster scanned device at a first resolution, said apparatus comprising:

10 a first processor for segmenting said output image into a plurality of sub-areas, and generating rendering instructions for sub-areas, said sub-areas having a second resolution and said second resolution being lower than said first resolution, wherein longitudinal sub-areas form bands;

15 a second processor for receiving said rendering instructions from said first processor, and rendering image data for said sub-areas at said second resolution, said second processor having a resolution limit sufficient for rendering said second resolution, but less than said first resolution;

memory means for storing sub-area image data into a band buffer;

20 colour converting means for colour converting said image data to form raster data suitable for said raster scanned device; and

data transfer means for transferring said raster data of a current band to said raster scanned device, upon all sub-areas of said current band being transferred into said band buffer.

### Brief Description of the Drawings

An embodiment of the present invention will now be described with reference to the drawings, in which:

Fig. 1 shows a block diagram of a printing apparatus;

Fig. 2 shows a flow diagram of a method of printing a high resolution output image displayed on the display device;

Fig. 3 shows a representation of an output page image broken up into tiles; and

Fig. 4 shows a flow diagram of a method of transferring printer codes interleaved with raster data to a print engine.

### Detailed Description including Best Mode

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same function(s) or operation(s), unless the contrary intention appears.

Fig. 1 shows a schematic block diagram of an apparatus 30 for printing a high-resolution image utilising a limited amount of memory. In the preferred embodiment, the apparatus 30 is a game console. Such a game console was not designed with a capability for printing high resolution outputs. The apparatus 30 comprises a computer module 31, a storage device 21, an input device such as control unit 22, and output devices including a print engine 20 and a display device 19. A user inputs command actions through the control unit 22 and the display device 19 displays command feedback and ancillary information. The print engine 20 may be that in a conventional inkjet printer, facsimile machine, plotter or copier, and prints pixels at fixed physical locations on an output medium such as paper. The higher resolution for printing is determined by the printer driver settings, such as resolution and paper size, of the print engine 20.

The computer module 31 typically includes at least one general purpose central processing unit (CPU) 10, a central memory unit 11, a direct memory access controller (DMAC) 12, a buffered graphics interface (BGIF) 13, and an input/output interface (I/O IF) 14. The components 10 to 14 of the computer module 31, communicate via an interconnected bus 24.

The I/O IF 14 also interfaces with an input/output (I/O) processor 15. The I/O processor 15 controls the operation of the peripheral devices, which includes the print engine 20, the storage device 21, control unit 22 and external peripherals (not illustrated) connected to interface buses 23, according to commands from the CPU 10. The I/O processor 15 also converts data into the appropriate format required by the peripheral devices 20, 21 and 22, and vice versa, and communicates with the CPU 10 and central memory 11 through the I/O IF 14. An I/O memory unit 16 is provided for dedicated use by the I/O processor 15.

The BGIF 13 is connected to a graphics renderer 17 which generates image data for the display device 19. A render memory unit 18 is also provided, connected to the graphics renderer 17 and for dedicated use by the graphics renderer 17. The graphics renderer 17 has limited capacity to process render instructions to form raster output data and the render memory unit 18 has limited capacity to store such raster output data. The capacities of the graphics renderer 17 and render memory unit 18 are sufficient to produce raster output data for the display device 19 and at the resolution required by the display device. Hence, these capacities are insufficient to process and store raster output data for a raster device at a higher resolution than that of the display device 19.

The DMAC 12 is programmed by the CPU 10 to perform direct data transfer to and from the central memory unit 11 and the other memory units, namely the render memory unit 18 and the I/O memory unit 16, without intervention by the CPU 10. The

I/O IF 14 manages the transfer of data between the central memory 11 and the I/O memory unit 16.

Typically, an application program is resident on the storage device 21. The application program is read from the storage device 21 and placed in the central memory unit 11, from where it is controlled in its execution by the CPU 10. The application program performs a number of graphics functions, including but not limited to display of three-dimensional models, user interaction with the three-dimensional models, display of images on the display device 19, loading of images from peripherals attached to the I/O processor 15 through the interface buses 23 etc. In the preferred implementation, a three dimensional model is used which is loaded from the storage device 21 and navigated by the user using the control unit 22.

The application program also includes functionality to print an image displayed on the display device 19. The image is displayed on the display device 19 by the graphics renderer 17, and the metrics for the displayed image are determined by the display device 19.

However, the printing facility is able to print a higher resolution image than the one displayed on the display device 19. Fig. 2 shows a flow diagram of a method 40 of generating a high-resolution output image for printing. The method 40 makes use of the graphics renderer 17 of the apparatus 30, which was never intended to be used for printing. The resolution of the image to be printed is also at a resolution higher than the capacities of the graphics renderer 17 and render memory unit 18. The steps of the method 40 are effected by instructions in the application program that are controlled by the CPU 10.

Printing is activated in step 41. During printing, the application program suspends controls of the display device 19, and assumes control of the BGIF 13, graphics renderer 17 and render memory unit 18. Referring also to Fig. 3 where a representation of an

output page 100 to be printed on the print engine 20 is shown. The method 40 includes the segmentation of the output page image 100 into an integral number of bands 101 to 104. The bands 101 to 104 have a width that is equal to the output page image width and have a height less than the page output image height. The bands 101 to 104 are further subdivided into an integral number of sub-areas or tiles 110, which in height are equal to the band height and have width less than the band width. In step 42 of the method 40, the CPU 10 calculates the dimensions of the tiles 110. This is done by considering the resolution limitations of the graphics renderer 17 and the render memory unit 18. Preferably, the smallest integral number of divisions of the output page 100 is found.

For example, if the graphics renderer 17 has a capacity of rendering a page with dimensions 640 pixels by 448 pixels to a NTSC display, and the higher resolution image for printing is a 600dpi resolution image, printed to an 8 inch by 6 inch page, hence 4800 pixels by 3600 pixels, then the output page 100 may be divided into an 10 by 10 array of tiles, or 100 tiles. Thus the output page 100 may be divided into tile dimensions of 480pixels by 360 pixels, with each tile having a resolution within the capacity of the graphics renderer 17 and the render memory unit 18.

Once tile dimensions has been determined, the CPU 10 modifies the graphics view transform matrix in step 43 to render a first tile. In step 44, the CPU 10 constructs render instructions into the central memory unit 11 and suitable for generation of a rendered image according to the new graphics view transform matrix. The graphics renderer 17 is activated by transferring the render instructions in the form of direct memory access (DMA) commands to the graphics renderer 17 in step 45. This causes the graphics renderer 17 to render a raster image of the tile into the render memory unit 18, and consequently also the display device 19.



Next, in step 46, the rendered raster image is transferred from render memory 18 to a band buffer in the central memory unit 11, again through use of the DMAC 12 and BGIF 13 in the form of DMA commands. This completes the rendering of the tile image.

Step 47 determines whether further tiles remain in the current band to be rendered.

- 5 If step 47 determines that tiles remain, then the method 40 returns to step 43 for rendering the next tile image. If step 47 determines that the all the tile for the current band has been rendered, then the band is marked as complete. Now that the band is completely rendered, it is queued for colour conversion processing in step 48.

- 10 Colour conversion is performed in step 48 by the CPU 10 on the completed band in the band buffer in the central memory unit 11. Typically, RGB pixel data suitable for displaying on the display device 19 is converted into CMYK pixel data, which is suitable for the print engine 20. In the preferred implementation, the tile rendering step 45 performed by the graphics renderer is performed concurrently with the band colour converting step 48 performed by the CPU 10.

- 15 Printer codes suitable for the printer engine 20 are generated in step 49, which are interleaved with the colour converted raster data of the band and transferred in step 50 to the I/O processor 15, and in particular to the I/O memory unit 16, through another DMA command. The operations within the I/O processor 15 of transferring the printer codes interleaved with raster data from a band to the print engine 20 are described below. Thus,  
20 each band is completely rendered before the band is colour converted, printer codes interleaved with the colour converted raster data and then transferred to the I/O processor.

- Step 51 determines whether any more bands remain for the output page image 100. If bands remain, then the method 40 returns to step 43 where the first tile of the next band is rendered. Alternatively, if step 51 determines that the last band of the output page  
25 image 100 has been sent to the I/O processor 15, then the method continues to step 52 where the view transform is returned to the settings prior to the print operation to again

display the full image in an resolution suitable for display device 19. The method 40 ends in step 53 where control of the display device 19 is returned to the graphics functions of the application program.

An application designer typically allocates an amount of memory of the central memory unit 11 for the band buffer. The band buffer memory is typically sufficient to store a whole optimally sized band. However, in a case where the band buffer is insufficient to store a whole optimally sized band, an integral number of scanlines for the band is determined, in step 42, that can be stored in the band buffer. For example, if the higher resolution image for printing is a 600dpi resolution image with 16 bits per pixel, printed to an 8 inch by 6 inch page, hence 4800 pixels by 3600 pixels, each band will have optimal dimensions 4800 pixels by 360 pixels. A band buffer of at least 3.5 Mbytes is required for such a print. In a case where only 40 Kbytes were allocated for the band buffer, the band buffer is sufficient for holding 2 scanlines of a rendered band. A band with dimensions 4800 pixels by 2 pixels, with 4 colour ink and 8 bits per ink to handle multiple inks or dot sizes, requires 38,400 bytes, which is within the 40Kbytes limit of the band buffer. Therefore, the band height, and consequently the tile size, is reduced to the determined number of scanlines.

If a graphics renderer 17 with a capacity of rendering 640 pixels by 448 pixels is used for rendering a 600dpi resolution image on a 8 inch by 5 inch page, hence 4800 pixels by 3000 pixels, then the output page 100 may be divided into an 8 by 300 array of tiles, each tile having dimensions of 600 pixels by 10 pixels. The smaller the height of the tile, in this case 10 pixels, the smaller the amount of memory required to store a band of raster data.

In a case where the band buffer is insufficient to hold one scanline, the band buffer is increased to a size sufficient to hold at least one scanline.

Fig. 4 shows a flow diagram of a method 60, performed by the I/O processor 15, of transferring the printer codes interleaved with raster data from the I/O memory unit 16 to the print engine 20. The method 60 operates concurrently with the method 40 performed by the CPU 10 and the graphics renderer 17. The method 60 starts in step 61 where instructions for performing the method 60 are loaded from the storage device 21. In step 62, the I/O processor 15 waits for I/O data, and in particular the printer codes interleaved with raster data, transferred from the CPU 10 to the I/O memory unit 16. Upon receipt of I/O data in step 63, the I/O data is serially transferred to the print engine 20 in step 64. The method returns to step 62 where the I/O processor 15 waits for more I/O data.

In a further implementation, the I/O data is transferred from print engine 20 back to the CPU 10 in error conditions.

In an alternative implementation, the generation of print code data in step 49 is not performed by the CPU in method 40, but rather performed by the I/O processor 15 in method 60. In particular, the generation of print code data is performed after print data in the form of raster data is received in step 63, and sent, interleaved with the raster data to the print engine 20 in step 64.

By generating rendered tiles one at a time, intermediate storage for page-rendering commands are minimised. In particular, by avoiding the need to generate display lists for multiple tiles on a page, memory usage is minimised. This also allows for a simplified graphics renderer to be used, one that does not use a display list architecture. For example, a high level graphics model renderer which renders directly to an output bitmap without the use of an intermediate low level graphics representation may be used to generate each tile in succession, that being achieved by successively modifying the high level graphics view transform to generate the correct high resolution tile for the

topologically mapped subarea on the output image as will be appreciated by those skilled in the art of computer graphics.

Another advantage is that a graphics renderer capable of limited output resolution may be employed to generate higher resolution images than the graphics renderer maximum output resolution. Accordingly, generalised rendering processors designed for display purposes other than printing may be employed to render tiles. Furthermore, software implemented raster image processors (RIPs) with limited output resolution capabilities may be employed to generate high-resolution images for printing.

For example, a RIP employing fixed point arithmetic will have limited output resolution ranges. However, through the use of the method 40, a high-resolution output image may be generated.

Yet another advantage is that a computer architecture containing a graphics renderer whose primary usage is other than generation of output image generation for printing may be employed to generate an output image for printing. For example, a general-purpose microprocessor may be coupled with a graphics renderer whose output bitmap is displayed on the display device. Such a graphics renderer may be used to perform render of tiles for generation of a high resolution output image whose resolution dimensions exceed the maximum output resolution capabilities of the rendering processor.

Yet another advantage is that a plurality of processors is employed to minimise output image generation time by partitioning the output image generation into a distinct number of stages which may each be performed by each processor within the confines of distinct memory storage areas so as to permit concurrent operation.

The foregoing describes only some implementations of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the implementations being illustrative and not restrictive. For

example, one or more of the steps of the preferred methods may be performed in parallel rather than sequentially.

Numerous advantages of the above-described implementations will be appreciated by those skilled in the art.

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	